

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A device for controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, said states determining the services offered by the object, said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of said memories (3, 4, 5) having a content defining a plurality of configurations, wherein said device comprises means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, and means for checking the content of the volatile memory, the data memories and the program memories of the portable electronic object as a function of the state transition to be effected, so that only some transitions are permitted amongst all the ~~possible~~ transitions between any two possible states of the portable electronic object.

2. (Canceled)

3. (Currently Amended) A device according to claim 1, ~~characterised in that~~ wherein the control means enable and/or inhibit a state transition, using a table (11) of permitted state transitions.

4. (Currently Amended) A device according to Claim 3, ~~characterised in that~~ wherein the control means comprise:

- in addition to the table (11) of permitted state transitions;
- a table (12) of the checks to be made per permitted state transition;
- and a check engine (9) using said tables.

5. (Currently Amended) A device according to Claim 3, ~~characterised in that~~ wherein the means of for controlling the transition from a first state to a second state of the portable electronic object comprise:

- an extension (16) to the table (11) of permitted state transitions.

6. (Currently Amended) A device according to Claim 4, ~~characterised in that~~ wherein the means of for controlling the transition from a first state to a second state of the portable electronic object comprise:

- an extension (16) to the table (11) of permitted state transitions;
- an extension (17) to the table (12) of checks to be made per permitted state transition;

and ~~in that~~ wherein the check engine (9) uses said table extensions (16, 17).

7. (Currently Amended) A device according to claim 1, ~~characterised in that~~ wherein the control means comprise means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object.

8. (Currently Amended) A device according to Claim 7 wherein said controlling means includes:

- an extension (16) to the table (14) of permitted state transitions;
- an extension (17) to the table (12) of checks to be made per permitted state transition;

and ~~in that~~ wherein the check engine (9) uses said table extensions (16, 17);
and

wherein said means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object comprise a table of actions (13) which can be used by the check engine (9).

9. (Currently Amended) A device according to Claim 8, ~~characterised in that~~ wherein the means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object comprise an extension (18) to the table (13) of actions which can be used by the check engine (9).

10. (Currently Amended) A portable electronic object having a processing unit (2), a volatile memory (3), program memories (4), data memories (5), and a device for controlling the life cycle of the object comprising means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, and means for checking the content of the volatile memory, the data memories and the program memories of the portable electronic object as a function of the state

transition to be effected, so that only some transitions are permitted amongst all the possible transitions between any two possible states of the portable electronic object.

11. (Currently Amended) A smart card having a processing unit (2), a volatile memory (3), program memories (4), data memories (5), and a device for controlling the life cycle of the object comprising means for controlling the transition from a first state to a second state of the ~~portable electronic object~~ smart card, including means for selectively enabling and/or inhibiting state transitions, and means for checking the content of the volatile memory, the data memories and the program memories of the smart card as a function of the state transition to be effected, so that only some transitions are permitted amongst all the possible transitions between two possible states of the smart card.

12. (Currently Amended) A method of controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, said states determining the services offered by the object, said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of said memories (3, 4, 5) having a content defining a plurality of configurations, said method being implemented, within the object, following a ~~state transition~~ request to transition from a current state to a new state, according to the following steps:

- a step ~~(51, 511, 528, 521)~~ of validation of the enabling of said request using means ~~of~~ for enabling and/or inhibiting state transitions, so that only certain

transitions are permitted amongst all the possible transitions between any two possible states of the object;

- a step of evaluating checks on the configuration of the object that are associated with a permitted transition; and

- a step ~~(57, 517, 527)~~ of ~~modifying the current~~ changing to the new state of the object if the requested transition is enabled ~~(51, 511, 528, 521)~~ and if said checks on the configuration of the object are satisfied.

13. (Currently Amended) A method according to Claim 12, ~~characterised in that it comprises~~ further comprising a step ~~(53, 513, 523)~~ of executing systematic actions associated with the requested transition.

14. (Canceled)

15. (Currently Amended) A method according to Claim 12, ~~characterized in that it comprises~~ further comprising a step ~~(56, 516, 526)~~ of executing positive actions performed if the requested transition is permitted ~~(51, 511, 528, 521)~~ and if the checks associated with the requested transition are satisfied ~~(54, 514, 524)~~.

16. (Currently Amended) A method according to claim 14, further including a step ~~(55, 515, 525)~~ of executing negative actions if the checks associated with the requested transition are not satisfied ~~(54, 514, 524)~~.

17. (Currently Amended) A method according to claim 12, further including a step ~~(56, 516, 526)~~ of executing positive actions ~~performed~~ if the requested transition is permitted ~~(51, 511, 528, 521)~~.

18. (Currently Amended) A method according to claim 12, implemented within the object, following a request for transition ~~from a first reference state to a second reference state~~, characterised in that, wherein the step (51) of validating the enabling of the said request ~~consists of~~ comprises analysing a table (41) of permitted transitions.

19. (Currently Amended) A method according to Claim 18, including the steps of:

- using an entry ~~(400, 401)~~, corresponding to the requested transition, in a table (43) of actions, and
- executing a ~~programme~~ program of actions (404) defined by the said entry.

20. (Currently Amended) A method according to claim 18, further including the step (52) of evaluating the checks associated with the requested transition comprising the steps of:

- using an entry (30) in a table (42) of checks, and
- executing a ~~programme~~ ~~(32)~~ program of checks defined by said entry.

21. (Currently Amended) A method according to claim 18 further including the step ~~(56)~~ of executing positive actions, if the requested transition is enabled ~~(51)~~

and if the checks associated with the requested transition are satisfied (54),

comprising the steps of:

- using an entry (400, 402), corresponding to the requested transition, in a table (13) of actions, and
- executing a ~~programme (405)~~ program of actions defined by the said entry.

22. (Currently Amended) A method according to claim 18 further including the step (55) of executing negative actions if the checks associated with the requested transition are not satisfied (54), comprising the steps of:

- using an entry (400, 403), corresponding to the requested transition, in the table (13) of actions, and
- executing a ~~programme (406)~~ program of actions defined by the said entry.

23. (Currently Amended) A method according to claim 18, further including the step (56) of executing positive actions if the requested transition is enabled (54), comprising the steps of:

- using an entry (400, 402), corresponding to the requested transition, in the table (13) of actions, and
- executing a ~~programme (405)~~ program of actions defined by the said entry.

24. (Currently Amended) A method ~~according to claim 12, implemented within the object~~ of controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, said states determining the services offered by the object, said object comprising a processing unit, a volatile

memory, program memories and data memories, each of said memories having a content defining a plurality of configurations, said method being implemented, within the object, following a request for transition from a first additive state to a second additive state, ~~characterised in that the step (511) of~~ comprising the steps of:

validating the enabling of said request ~~consists of~~ by analysing an extension (16) to a table (11) of permitted transitions, so that only certain transitions are permitted amongst all the transitions between any two possible states of the object;
and

changing the state of the object to the second additive state if the requested transition is enabled.

25. (Currently Amended) A method according to Claim 24, further including the step ~~(513)~~ of executing systematic actions comprising the steps of:

- using an entry ~~(407, 408)~~, corresponding to the requested transition, in an extension (18) to a table (13) of actions, and
- executing a ~~programme~~ program of actions (411) defined by said entry.

26. (Currently Amended) A method according to claim 24, further including the step ~~(512)~~ of evaluating the checks associated with the requested transition, comprising the steps of:

- using an entry ~~(33)~~ in an extension (17) to a table (12) of checks, and
- executing a ~~programme~~ ~~(35)~~ program of checks defined by said entry.

27. (Currently Amended) A method according to claim 24 further including the step ~~(516)~~ of executing positive actions if the requested transition is enabled ~~(511)~~ and if the checks associated with the requested transition are satisfied ~~(514)~~, comprising the steps of:

- using an entry ~~(407, 409)~~, corresponding to the requested transition, in an extension ~~(18)~~ to a table ~~(13)~~ of actions, and
- executing a ~~programme (412)~~ program of actions defined by said entry.

28. (Currently Amended) A method according to claim 24, further including the step ~~(515)~~ of executing negative actions if the checks associated with the requested transition are not satisfied ~~(514)~~, comprising the steps of:

- using an entry ~~(407, 410)~~, corresponding to the requested transition, in an extension ~~(18)~~ to a table ~~(13)~~ of actions, and
- executing a ~~programme (413)~~ program of actions defined by ~~the~~ said entry.

29. (Currently Amended) A method according to claim 24, further including the step ~~(516)~~ of executing positive actions if the requested transition is enabled ~~(511)~~, of:

- using an entry ~~(407, 409)~~, corresponding to the requested transition, in an extension ~~(18)~~ to the table ~~(13)~~ of actions, and
- executing a ~~programme (412)~~ program of actions defined by said entry.

30. (Currently Amended) A method ~~according to claim 12, implemented within the object~~ of controlling the life cycle of a portable electronic object, the life

cycle being determined by a succession of state transitions, said states determining the services offered by the object, said object comprising a processing unit, a volatile memory, program memories and data memories, each of said memories having a content defining a plurality of configurations, said method being implemented, within the object, following a request for transition from a reference state to an additive state, characterised in that the step (528, 521) of comprising the steps of:

- validating the enabling of said request consists of by:

- validating (528) the enabling of a transition from said reference state to an additive state, whilst analysing a table (41) of permitted transitions;

- validating (521) the enabling of a transition from said reference state to said additive state, whilst analysing an extension (46) to the table (41) of permitted transitions, so that only certain transitions are permitted amongst all the transitions between any two possible states of the object; and

changing the state of the object to the additive state if the requested transition is enabled.

31. (Currently Amended) A method according to Claim 30, further including the step (513) of executing systematic actions comprising the steps of:

- using an entry (407, 408) corresponding to the requested transition, in an extension (48) to a table (43) of actions, and

- executing a programme program of actions (411) defined by said entry.

32. (Currently Amended) A method according to claim 30, further including the step ~~(522)~~ of evaluating the checks associated with the requested transition comprising the steps of:

- using an entry ~~(33)~~ in an extension ~~(17)~~ to a table ~~(12)~~ of checks, and
- executing a programme ~~(35)~~ of checks defined by said entry.

33. (Currently Amended) A method according to claim 30, further including the step ~~(526)~~ of executing positive actions if the requested transition is enabled ~~(528, 521)~~ and if the checks associated with the requested transition are satisfied ~~(524)~~, comprising the steps of:

- using an entry ~~(407, 409)~~, corresponding to the requested transition, in an extension ~~(18)~~ to a table ~~(13)~~ of actions, and
- executing a programme ~~(412)~~ program of actions defined by said entry.

34. (Currently Amended) A method according to claim 30, further including the step ~~(525)~~ of executing negative actions if the checks associated with the requested transition are not satisfied ~~(524)~~, comprising the steps of:

- using an entry ~~(407, 410)~~, corresponding to the requested transition, in an extension ~~(18)~~ to a table ~~(13)~~ of actions, and
- executing a programme ~~(413)~~ program of actions defined by said entry.

35. (Currently Amended) A method according to claim 30, further including the step ~~(526)~~ of executing positive actions if the requested transition is enabled ~~(528, 521)~~, of:

- using an entry ~~(407, 409)~~, corresponding to the requested transition, in an extension ~~(18)~~ to the table ~~(13)~~ of actions, and
- executing a program ~~(412)~~ of actions defined by said entry.

36. (Previously Presented) A method according to claim 12, wherein said method does not enable the crossover of a state transition, from an additive state to a reference state.

37. (New) The method according to claim 1, wherein said checking means determines whether said memories contain data that is invalid for the transition to be effected.

38. (New) The method according to claim 12, wherein said evaluation step comprises checking whether said memories have a predetermined configuration associated with the transition from said current state to said new state.